# Flexible Photogrammetric Computations using Modular Bundle Adjustment

The chain rule and the collinearity equations

Niclas Börlin (1,*), Arnadi Murtiyoso (2), Pierre Grussenmeyer (2), Fabio Menna (3,4), Erica Nocerino (5,6)

(1) Department of Computing Science, Umeå University, SE-90187 Umeå, Sweden, niclas.borlin@cs.umu.se

(2) Photogrammetry and Geomatics Group, ICube Laboratory UMR 7357, INSA Strasbourg, France, (arnadi.murtiyoso, pierre.grussenmeyer)@insa-strasbourg.fr

(3) 3D Optical Metrology (3DOM) unit, Bruno Kessler Foundation (FBK), Trento, Italy, fmenna@fbk.eu

(4) Currently at COMEX SA – Innovation Department, COMEX, 36 bd de l'Océan, CS 80143, 13275 Marseille, France, f.menna@comex.fr

(5) LIS, I&M Team, Aix-Marseille Universite, Polytech Luminy, 13288 Marseille, France, erica.nocerino@univ-amu.fr

(6) Institute of Theoretical Physics, ETH Zurich, 8093 Zurich, Switzerland, erican@phys.ethz.ch

(7) (*) Corresponding author

Abstract

The main purpose of this paper is to show that photogrammetric bundle adjustment computations can be sequentially organized into modules. Furthermore, the chain rule can be used to simplify the computation of the analytical Jacobians needed by the adjustment. Novel projection models can be flexibly evaluated by inserting, modifying, or swapping the order of selected modules.

As a proof of concept, two variants of the pin-hole projection model with Brown lens distortion were implemented in the open-source Damped Bundle Adjustment Toolbox (DBAT) and applied to simulated and calibration data for a non-conventional lens system. The results show a significant difference for the simulated, error-free, data but not for the real calibration data.

The current flexible implementation incurs a performance loss. However, in cases where flexibility is more important, the modular formulation should be a useful tool to investigate novel sensors, data processing techniques, and refractive models.

# 1 Introduction

## 1.1 Background

In Photogrammetry and Computer Vision, Bundle Adjustment (BA) is a fine-tuning estimation process where some or all parameters that describe a scene are estimated simultaneously. Bundle adjustment was introduced to photogrammetry some sixty years ago (Brown, 1958). The technique became popular in the Computer Vision community after the paper by Triggs et al. (2000).

Photogrammetric BA uses the collinearity equations to describe the world-to-camera projection. The typical presentation of the function and its linearization has been as a single, compact computation, see, e.g., Kraus (1993, Sec. 5.3.2), Wolf and Dewitt (2000, App. D-4), Mikhail et al. (2001, App. C.3), and Luhmann et al. (2014, Sec. 4.2.2). In contrast, the standard presentation of the world-to-camera projection in the Computer Vision community has been as a sequence of small operations (Tsai, 1987) with the Jacobians described using the chain rule (Brown and Lowe, 2003; Fusiello, 2013). In both presentations, the lens distortion has been treated separately. Although both research communities refer to the same Brown (1971) paper, the details of the application has differed.

Since its introduction, BA has evolved to be a central algorithm in photogrammetry and computer vision (Brown, 1976; Triggs et al., 2000; Luhmann et al., 2014). Originally, the procedure was used by a small research community that understood the algorithm in detail. With a broadening user base, the fraction of users with an intimate understanding of the inner workings of BA has declined. This development has further increased with the inclusion of BA into commercial, black-box, software.

## 1.2 Related work

The Damped Bundle Adjustment Toolbox (DBAT) was originally developed to illustrate the behaviour of different damping schemes known from non-linear optimization (Börlin and Grussenmeyer, 2013; 2014). Later work has focused on the use to validate commercial black-box photogrammetric software, e.g., PhotoModeler and Photoscan (Börlin and Grussenmeyer, 2016), especially to provide

detailed diagnostics of the photogrammetric network (Dall'Asta et al., 2015; Murtiyoso et al., 2017; Börlin et al., 2018). Recently, the differential observation weighting capabilities of DBAT was used by Menna et al. (2018) to mitigate peripheral image quality degradation in underwater photogrammetry.

The chain rule has been used to simplify geometrical computations in e.g., computer vision (Brown and Lowe, 2003; Fusiello, 2013), computer graphics (Barr, 1984; Piponi, 2004), imaging (Gallego and Yezzi, 2015), robotics (Koditschek and Rimon, 1990), and registration of TLS point clouds (Rabbani et al., 2007).

In photogrammetry, the chain rule has been used for, e.g., modelling of reflected ray paths (Rupnik et al., 2015) and relative pose estimation (Cheng et al., 2017). To the authors' knowledge, the chain rule has not been used to express the photogrammetric world-to-camera projection model, including lens distortion, before.

### 1.3 Aim

The aim of this paper is to apply the sequential computation to the whole photogrammetric projection model, including lens distortion, and to use the chain rule to modularize the computations. Furthermore, to illustrate the flexibility of the modular approach, a non-conventional optical system – a tilt-shift lens – will be investigated.

A first version of this approach was presented in Börlin et al. (2018). This paper is an extended version with an added emphasis on performance vs. flexibility.

## 2 Disassembling collinearity

### 2.1 The collinearity equations

The mathematical problem that is solved by the BA process includes a residual between two components. One residual component is the simulated projection of an object point according to

the camera model at hand. The other component is computed from the corresponding image measurement. As a typical presentation, consider the collinearity equations as presented by Luhmann et al. (2014, equation (4.94)):

$$\begin{aligned} x' &= x'_0 + z' \frac{r_{11}(X - X_0) + r_{21}(Y - Y_0) + r_{31}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} + \Delta x', \\ y' &= \mathrm{y}'_0 + z' \frac{r_{12}(X - X_0) + r_{22}(Y - Y_0) + r_{32}(Z - Z_0)}{r_{13}(X - X_0) + r_{23}(Y - Y_0) + r_{33}(Z - Z_0)} + \Delta \mathrm{y}'. \end{aligned} \tag{1}$$

Ignoring $\Delta \mathrm{x}'$ and $\Delta \mathrm{y}'$, equation (1) describes the projection of an object point at $(X, Y, Z)$ through a pinhole camera with principal point $(x'_0, y'_0)$ and camera constant $z'$ placed at $(X_0, Y_0, Z_0)$ and with an orientation given by the matrix $R$.

Another requirement for the bundle adjustment process is the linearization of equation (1) with respect to any parameter that is to be estimated, see, e.g., Kraus (1993, Sec. 5.3.2), Wolf and Dewitt (2000, App. D-4), or Mikhail et al. (2001, App. C.3). Some partial derivatives of equation (1) are presented by Luhmann et al. (2014, equation (4.96)) as

$$\begin{aligned} \frac{\partial x'}{\partial X} &= -\frac{z'}{N^2}(r_{13}k_X - r_{11}N), & \frac{\partial y'}{\partial X} &= -\frac{z'}{N^2}(r_{13}k_Y - r_{12}N), \\ \frac{\partial x'}{\partial Y} &= -\frac{z'}{N^2}(r_{23}k_X - r_{21}N), & \frac{\partial y'}{\partial Y} &= -\frac{z'}{N^2}(r_{23}k_Y - r_{22}N), \\ \frac{\partial x'}{\partial Z} &= -\frac{z'}{N^2}(r_{33}k_X - r_{31}N), & \frac{\partial y'}{\partial Z} &= -\frac{z'}{N^2}(r_{33}k_Y - r_{32}N), \end{aligned} \tag{2}$$

where $k_X$ and $k_Y$ are the respective numerators of equation (1), and $N$ is the denominator.

## 2.2 Basic functions

If we group the parameters as

$$\begin{aligned} p &= \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \quad p_0 = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix}, \quad q = \begin{pmatrix} k_X \\ k_Y \\ N \end{pmatrix}, \\ u &= \begin{pmatrix} x' \\ y' \end{pmatrix}, \quad u_0 = \begin{pmatrix} x'_0 \\ y'_0 \end{pmatrix}, \quad \Delta u = \begin{pmatrix} \Delta x' \\ \Delta y' \end{pmatrix}, \end{aligned} \tag{3}$$

and introduce the basic functions

$$\begin{array}{rcll} T_3(p, p_0) & = & p + p_0, & \text{3D translation} \\ L(R, v) & = & Rv, & \text{3D rotation} \\ H(q) & = & \frac{1}{q_3}\begin{pmatrix} q_1 \\ q_2 \end{pmatrix}, & \text{Pinhole projection} \\ S(c, m) & = & cm, & \text{Scaling} \\ T_2(n, u_0) & = & n + u_0, & \text{2D translation} \end{array} \tag{4}$$

where $c$ is scalar and $m$ and $n$ are 2-vectors, we may write equation (1) as the following composition:

$$u = g(p, p_0, R, z', u_0) \equiv T_2\left(S\left(z', H\left(L\left(R^T, T_3(p, -p_0)\right)\right)\right), u_0\right). \tag{5}$$

In the definition of the functions in equation (4), generality has been favoured over typical usage to simplify future extensions. For instance, the function $L(R, v)$ describes an arbitrary 3D linear transformation whereas the collinearity equations assumes $R$ to be a rotation matrix. Similarly, the translation $T_3(p, p_0)$ is defined with a positive sign on $p_0$ whereas the collinearity equations use $T_3(p, -p_0)$.

The algorithm corresponding to equation (5) is shown in Algorithm 1 with a graphical representation in Figure 1.

## 2.3 Linearization

The linearization of function compositions is governed by the chain rule, see, e.g., Magnus and Neudecker (2007). For example, the Jacobian of the projection function $g$ in equation (5) with respect to the object point coordinates $p$ is the matrix product of five Jacobians

$$\left[\frac{dg}{dp}\right] = \left[\frac{dT_2}{dn}\right]_{n=S(\dots)}\left[\frac{dS}{dm}\right]_{m=H(\dots)}\left[\frac{dH}{dq}\right]_{q=L(\dots)}\left[\frac{dL}{dv}\right]_{v=T_3(\dots)}\left[\frac{dT_3}{dp}\right], \tag{6}$$

where the brackets indicate a matrix and the subscript indicate the point at which the respective Jacobians are evaluated. A detailed inspection of equation (6) reveals that it is identical to equation (2).

## 2.4 A function as a module

If we require that each basic function of equation (4) can compute both the function value and the Jacobians with respect to all its parameters, we may similarly extend Algorithm 1 to compute both its function value and its Jacobians. The extended Algorithm 2 illustrates the computation of equation (5) and the Jacobian of equation (6). In this way we can modularize the residual computations and use the modules as building blocks. Furthermore, as each module is self-contained, it is possible to validate the analytical Jacobian of each module independently.

## 2.5 Other photogrammetric modules

In addition to the functions listed in equation (4), we add the computation of the rotation matrix $R$ from the $\omega - \phi - \kappa$ Euler angles (Förstner & Wrobel, 2004, equations (2.128)-(2.130))

$$R(\omega, \phi, \kappa) = R_3(\kappa)R_2(\phi)R_1(\omega), \tag{7}$$

the Brown (1971) lens distortion model

$$D(u, K, P) = u + d_r(u, K) + d_t(u, P), \tag{8}$$

and the 2D affine transformation

$$A_2(u, b) = \begin{pmatrix} 1 + b_1 & b_2 \\ 0 & 1 \end{pmatrix} u. \tag{9}$$

In equation (8), the vectors $K$ and $P$ contain the radial and tangential distortion coefficients, respectively. In equation (9), the scalar $b_1$ controls the aspect ratio and the scalar $b_2$ controls the skew. For more details and derivation of the Jacobians, see (Börlin et .al (2018, App. B).

# 3 Reassembling collinearity

## 3.1 The Damped Bundle Adjustment Toolbox (DBAT)

The modular technique has been used in the open-source Damped Bundle Adjustment Toolbox (DBAT)[1] package for Matlab (Börlin and Grussenmeyer, 2013; 2016). For instance, the modules presented in equation (4) and (7)-(9) were combined differently to implement two bundle pipelines that used different adaptations of the Brown (1971) lens distortion model. In the Photogrammetric formulation, the Brown polynomials are used to "correct" for the effect of lens distortion on the measured image coordinates. In contrast, the formulation largely adopted by the Computer Vision community uses the same polynomials to "add" lens distortion to the ideal projection of object points (Tsai, 1987; Heikkilä and Silvén, 1997; Zhang, 2000; Drap and Lefévre, 2016). The two pipelines are visually contrasted in figures Figure 2 and Figure 3.

DBAT has the possibility to post-process files from two popular software: EOS Photomodeler and Agisoft Photoscan. DBAT version 0.8 was used in this paper.

## 3.2 Extended pinhole models

For this paper, we have used the modular technique used in DBAT to implement two extensions of the photogrammetric pipeline of Figure 2. The reference implementation (called Model 2) has no affinity. In Model 3, the affinity is applied *before* lens distortion. In Model 4, the affinity is applied *after*. The corresponding functions are:

$$\begin{aligned} r_2 &= D(T_2(S(s,u),-u_0),K,P), \\ r_3 &= D(A_2(T_2(S(s,u),-u_0),b),K,P), \\ r_4 &= A_2(D(T_2(S(s,u),-u_0),K,P),b), \end{aligned} \tag{10}$$

where $s$ is the pixel size and $u$ the measured image coordinates. The models are visually illustrated in Figure 4. The difference in the computation of the Jacobian with respect to the principal point $u_0$

[1] https://github.com/niclasborlin/dbat

was limited to inserting the Jacobian $\left[\frac{dA_2}{du}\right]$ at the proper place in the matrix product (in addition to a change in the evaluation points):

$$\begin{aligned}\left[\frac{dr_2}{du_0}\right] &= -\left[\frac{dD}{du}\right]_{u=T_2(\dots)}\left[\frac{dT_2}{du_0}\right], \\ \left[\frac{dr_3}{du_0}\right] &= -\left[\frac{dD}{du}\right]_{u=A_2(\dots)}\left[\frac{dA_2}{du}\right]_{u=T_2(\dots)}\left[\frac{dT_2}{du_0}\right], \\ \left[\frac{dr_4}{du_0}\right] &= -\left[\frac{dA_2}{du}\right]_{u=D(\dots)}\left[\frac{dD}{du}\right]_{u=T_2(\dots)}\left[\frac{dT_2}{du_0}\right],\end{aligned} \tag{11}$$

### 3.3 Image-variant estimation

A recent addition to DBAT allows IO parameters to be declared as block-invariant or image-variant (Moniwa, 1981) on an individual basis. Thus, some parameters can be the same for the full block whereas others can be individual for an image.

## 4 Experiments and results

### 4.1 Test case: The tilt-shift lens

A tilt-shift lens is a non-conventional optical system that allows the following lens movements (Ray, 2002) (Figure 5):

- a *tilt*, i.e., a rotation of the optical axis about either the exit pupil or the centre of the sensor plane,
- a *shift*, i.e., a translation of the optical axis, and
- a *rotation*, i.e., a rotation about the optical axis.

The projection of a tilt-shift camera lens system is not completely modelled by the classic photogrammetric projection model (pin-hole with Brown lens distortion) available in regular photogrammetric software, including DBAT. In order to illustrate the modularity of DBAT, we chose to investigate whether the ordering of the affine and lens distortion steps, described in Section 3.2, could improve the projection model.

One simulation experiment, two calibration experiments, and one performance experiment were designed. The purpose of the simulation experiment was to investigate whether the variants in the application of the affinity had any measurable effect. The purpose of the calibration experiments was to investigate whether any of the affinity variants could improve an actual calibration of the tilt-shift lens, with or without image-variant camera parameters. Finally, the purpose of the performance experiment was to investigate the relative performance of the DBAT models.

## 4.2 Simulation experiment

The first experiment was constructed to investigate the effect, if any, of the difference in the affine-lens distortion ordering. Two sets of synthetic data were generated, where simulated error-free image observations were generated by back-projection of known 3D object coordinates and with known exterior orientation parameters (EO), and camera calibration parameters (IO) of a strong self-calibration network. The network consisted of 24 cameras at varying roll angles. About 100 targets were simulated in a 3D configuration with a largest dimension of 1000 mm (Figure 6).

### 4.2.1 Generation of synthetic data

The following algorithms were used to simulate measurements generated by Model 3 and Model 4. Note that the sequence of the steps below is reversed compared to Figure 4 as we are building image observations.

#### *4.2.1.1 Model 3*

1. Apply the collinearity equations (3D translation, 3D rotation, pinhole projection and optical scaling).
2. Introduce lens distortion (iterative, [mm]).
3. Convert from mm to pixels using a non-square pixel size ($b_1 = 0.01218$).
4. Introduce the principal point [pixels].

#### *4.2.1.2 Model 4*

1. Apply the collinearity equations (3D translation, 3D rotation, pinhole projection and optical scaling).
2. Apply an affine transformation to the image coordinates ($b_1 = 0.01218$).
3. Introduce lens distortion (iterative, [mm]).
4. Convert from mm to pixels using a square pixel size.
5. Introduce the principal point [pixel].

Both simulations used a skew (shear) parameter of $b_2 = 0$. The algorithms were implemented in software developed in-house at FBK and not by DBAT.

### 4.2.2 Analysis of synthetic data

Each synthetic data set was analysed by a self-calibration bundle adjustment using DBAT models 2, 3, and 4. No control points were used. Instead, the datum problem was solved by fixing seven EO parameters. The prior weight for the image observations corresponded to a sigma of 0.1 pixels. The following parameters were estimated; the camera constant $c$, the principal point $(x_0, y_0)$, the radial distortion parameters $K_1 - K_3$, and the tangential distortion parameters $P_1 - P_2$. For models 3 and 4, the affine parameters $b_1 - b_2$ were also estimated. The quality of each analysis was evaluated in image space ($\sigma_0$ and 2D image point RMS) and object space (3D RMSE between the true and estimated OP coordinates). The results are given in Table 1. When the correct estimation model was used, the simulated $b_1$ value was recovered to the number of available decimals and the internal and external residuals were effectively zero. When the wrong model was used, the residuals were significantly higher.

## 4.3 Camera calibration

In the second experiment, models 2, 3 and 4 were used to analyse calibration data for a camera with a tilt-shift lens. The data set from (Nocerino et al. (2016) was used for the calibration. The dataset was acquired in a controlled environment, with stable environmental conditions, consequently reducing the influences of uncontrolled factors on the calibration. Furthermore, high accuracy reference coordinates were available: A point precision vector length of 2.4 µm and an RMSE with respect to another photogrammetric system of 6 µm. The calibration target was a 3D photogrammetric calibration test object (Figure 7) with about 170 coded targets with a largest dimension of 900 mm. The used camera was a Nikon D750 full-frame DSLR camera with a PC-E Micro NIKKOR 45mm f/2.8D ED tilt-shift lens (Figure 5) in two different configurations:

- NORMAL: The normal configuration where neither tilt nor shift was applied.

- TILTED: The lens was tilted in the vertical plane by 4 degrees.

A high redundancy photogrammetric camera network was realized, consisting of up to 48 convergent images with a diversity of camera roll angles to enhance the determinability of the IO parameters (Fraser, 2001). The ground sample distance (GSD) was about 0.13 mm. The image measurements were performed by PhotoModeler Scanner v.6 and exported as a text file that was imported to DBAT.

The NORMAL data set was analysed and used as a reference for the processing of the TILTED data set. The TILTED data set was analysed by a self-calibration bundle adjustment in DBAT using models 2, 3 or 4. The datum and prior weights were the same as in the synthetic experiment. Furthermore, the parameters $b_1$, $b_2$, $P_1$, and $P_2$ were individually included or excluded from the estimation ($P_1$ and $P_2$ were always estimated together). The quality of the estimation was evaluated as in the synthetic experiment with the results of the NORMAL data set used as the reference. The results are given in Table 2.

From Table 2, we see that the difference between models 3 and 4 on this data is small. The internal and external residuals were small when the aspect parameter $b_1$ and the decentering distortion parameters $P_1$-$P_2$ were included in the estimation. In those cases, the 3D RMSE was about 0.09 GSD and the estimated $b_1$ value was about 0.0019. This value is equal to the value in Nocerino et al. (2016, Table 3, column T-4S0R90) that was estimated by other software. In contrast, the inclusion of $b_2$ resulted in an estimated value of $b_2$ was about $10^{-5}$ and had a negligible effect on the residuals.

## 4.4 Image-variant calibration

The best $b_2$-free setup for model 3 from the second experiment was used as the reference for the third experiment. Three different sets of IO parameters were allowed to be image-variant. The first set consisted of the principal distance and point. The second set added the radial and tangential lens distortion parameter. The final set consisted of all IO parameters, including the aspect parameter $b_1$. In order to avoid too weak networks, prior observations of the image-variant parameters were

added Tecklenburg et al. (2001). As the prior values, the posterior estimated values from the reference experiment were used. The prior weights were chosen to correspond to twice the posterior standard deviation from the reference experiment. The results are given in Table 3.

The internal assessment suggests a slight improvement in accuracy. However, the comparison with the reference 3D coordinates instead suggests that the accuracy slightly worsens for all image-variant experiments.

### 4.5 Performance

To investigate the relative performance between the models, the St-Pierre-le-Jeune data set from Murtiyoso et al. (2017) was used. The data set consisted of 239 images and about 18000 object points. DBAT model 1, a previous, more optimized implementation of Model 2, was used as a performance reference. A self-calibration bundle adjustment process was applied to the St-Pierre data set. The timings were performed using an HP EliteDesk 800 with Quad Intel Core

i7-4790 CPU at 3.60GHz with 32 GB RAM running Matlab R2018b under Ubuntu Linux 14.04.5. In all cases, the models required the same number of iterations and produced the same results to the reported number of digits. The average bundle execution time over five repeated runs was about 13 seconds for Model 1. The corresponding times for models 2-4 were 53-56% higher.

## 5 Discussion and conclusion

The primary goal of this paper was to apply the sequential presentation to the whole photogrammetric projection model, including lens distortion, and to use the chain rule to modularize the computations. Previous presentations have left out the camera-internal

operations, notably lens distortion. The presentation in this paper does show that the residual computations used by bundle adjustment can be split into modules. If each module is responsible for computing Jacobians with respect to each parameter, in addition to the function value proper, the chain rule can be used to combine the analytical Jacobians of the simple operations into analytical

Jacobians of complex expressions. Furthermore, the presentation illustrates how the Brown (1971) lens distortion model is treated differently by the photogrammetry and computer vision communities.

The tilt-shift lens is a complex design whose projection model is not yet rigorously supported by regular photogrammetric software, including DBAT. In a previous paper, it was found that some of the deviation of the tilt-shift lens from the classic photogrammetric projection model (pin-hole with Brown lens distortion) could be mitigated by including the affine distortion parameter $b_1$ in the estimation (Nocerino et al., 2016). The modularity of DBAT presented an opportunity to investigate further modifications of the projection model. As such, two models with different placement of the affine transformation compared to lens distortion were implemented with minimal effort.

An experiment on synthetic image observations verified that the relative placement of the affine transformation and lens distortion can be detected by DBAT on synthetic, error-free data. A calibration experiment was constructed to test the hypothesis that either ordering could explain more of the tilt-shift distortion of a real tilt-shift lens. Furthermore, an image-variant bundle adjustment (Moniwa, 1981; Tecklenburg et al., 2001) was tested to verify if it would be possible to better model the distortions introduced by gravity effects and the tilt angle. The results did not support either hypothesis. Compared to other, possibly unmodelled effects, the ordering does not significantly contribute to the observed residuals and the addition of image-variant camera parameters did not improve the results. Instead, the results support the conclusion in Nocerino et al. (2016) that the aspect parameter $b_1$ and the decenter parameters $P_1$-$P_2$ are significant but the skew parameter $b_2$ is not.

From the results, we may conclude that the standard pinhole camera model with Brown lens distortion cannot fit the tilted data better, even with the image-variant approach. However, if an RMSE of 0.09 times the GSD is considered satisfactory, standard photogrammetric formulations and

software can be used in 3D modelling application that require the tilt-shift lens system. We believe that to improve on those results, a proper mathematical projection model of the tilt-shift lens is required. In relation to the main goals of this paper, we conclude that the modularity of DBAT can indeed be used to test novel projection models with minimal implementation effort.

The performance results in this paper show an increase of about 50% in computation time on a medium-sized problem for the flexible implementation compared to the original implementation in DBAT. A potential explanation is if the computation of the composed Jacobian as a chained multiplication of simpler Jacobians is inefficient. Another possible explanation is if some of the computations that depend on the same parameters are distributed over multiple sub-expressions. In that case, the expressions may need to be jointly considered in order to be fully optimized. If true, either case would highlight a limitation of the chain rule approach.

The reported increase in execution time would hardly be acceptable for production code, where speed is essential. However, the driving force behind the development of DBAT has been to provide an open-source, statistically rigorous, reference implementation of Bundle Adjustment. Performance has been important, but never the primary focus. Furthermore, the code for the original projection model has gone through several optimization iterations whereas the flexible chain rule code has not. It remains to be seen how much performance can be recovered in the future.

In some cases, flexibility is more important than performance. The flexibility of the chain rule formulation should allow researchers to investigate novel and non-standard sensors or data processing techniques. Furthermore, several new photogrammetric applications, including underwater applications, need refined refractive models. In teaching, the flexibility should allow students to experiment with both existing and novel computations.

In summary, the presentation and results in this paper show that it is possible to use the chain rule to modularize the bundle adjustment computations, including lens distortion. Currently, there is a

performance loss associated with the modular implementation. In cases where flexibility is more important than performance, we argue that DBAT with the chain rule formulation can be a useful tool to investigate novel sensors, data processing techniques, and refractive models.

Future work include optimization of the flexible models, complementing the pin-hole module with a fish-eye module, complementing the omega-phi-kappa rotation model by, e.g. azimuth-tilt-swing, and extensions to refractive models to handle the tilt-shift lens or multimedia projection models in, e.g., underwater photogrammetry.

# 6 References

*Barr A. H. 1984. Global and local deformations of solid primitives, ACM SIGGRAPH Computer Graphics 18(3):21-30. DOI: 10.1145/964965.808573.*

*Börlin N. and P. Grussenmeyer 2013. Bundle adjustment with and without damping, Photogrammetric Record 28(144):396-415. DOI: 10.1111/phor.12037.*

*Börlin N. and P. Grussenmeyer 2014. Camera calibration using the damped bundle adjustment toolbox, ISPRS Annals of the Photogrammetry, Remote Sensing, and Spatial Information Sciences II(5):89-96. DOI: 10.5194/isprsannals-II-5-89-2014.*

*Börlin N. and P. Grussenmeyer 2016. External verification of the bundle adjustment in photogrammetric software using the damped bundle adjustment toolbox, International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences XLI(B5):7-14. DOI: 10.5194/isprs-archives-XLI-B5-7-2016.*

*Börlin N. A. Murtiyoso P. Grussenmeyer F. Menna and E. Nocerino 2018. Modular bundle adjustment for photogrammeric computations, International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences XLII(2):133-140. DOI: 10.5194/isprs-archives-XLII-2-133-2018.*

*Brown D.C. 1958. A solution to the general problem of multiple station analytical stereotriangulation RCA Data reduction technical report 43 Ballistic Research Laboratories Aberdeen Proving Ground Maryland USA.*

*Brown D.C. 1971. Close-range camera calibration, Photogrammetric Engineering 37(8):855-866.*

*Brown D.C. 1976. The bundle adjustment - progress and prospects, International Archives of Photogrammetry and Remote Sensing 21(3):1-33.*

*Brown M. and D.G. Lowe 2003. Recognising panoramas, 9th IEEE International Conference on Computer Vision (ICCV 2003) 14-17 October 2003 Nice France pp. 1218-1227. DOI: 10.1109/ICCV.2003.1238630.*

*Cheng C. X. Hao and J. Li 2017. Relative camera pose estimation method using optimization on the manifold, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII(1/W1):41-46. DOI: 10.5194/isprs-archives-XLII-1-W1-41-2017.*

*Dall'Asta E. K. Thoeni M. Santise G. Forlani A. Giacomini and R. Roncella 2015. Network design and quality checks in automatic orientation of close-range photogrammetric blocks, Sensors 15(4):7985-8008. DOI: 10.3390/s150407985.*

*Drap P. and J. Lefévre 2016. An exact formula for calculating inverse radial lens distortions, Sensors 16(6):807. DOI: 10.3390/s16060807.*

*Förstner W. and B. Wrobel 2004. Mathematical concepts in photogrammetry Manual of Photogrammetry 5th ed. (C. McGlone E. Mikhail and J. Bethel editors) ASPRS Bethesda Maryland pp. 15-180.*

*Fraser C.S. 2001. Photogrammetric camera component calibration: A review of analytical techniques Calibration and Orientation of Cameras in Computer Vision (A. Gruen and T.S. Huang editors), Springer Heidelberg Germany pp. 95-121.*

*Fusiello A. 2013. Visione computazionale. Tecniche di ricostruzione tridimensionale. Franco Angeli Milano Italy 340 p.*

*Gallego G. and A. Yezzi 2015. A compact formula for the derivative of a 3-d rotation in exponential coordinates, Journal of Mathematical Imaging and Vision 51(3):378-384. DOI: 10.1007/s10851-014-0528-x.*

*Heikkilä J. and O. Silvén 1997. A four-step camera calibration procedure with implicit image correction, Proceedings of IEEE Computer Society Conference on Computer Vision and*

*Pattern Recognition (CVPR) 17-19 June 1997 San Juan Puerto Rico USA pp. 1106-1112. DOI: 10.1109/CVPR.1997.609468.*

*Koditschek D.E. and E. Rimon 1990. Robot navigation functions on manifolds with boundary, Advances in Applied Mathematics 11(4):412-442. DOI: 10.1016/0196-8858(90)90017-S.*

*Kraus K. 1993. Photogrammetry 4th edition Dümmler Verlag Bonn Germany 397 p.*

*Luhmann T. S. Robson S. Kyle and J. Boehm 2014. Close-range photogrammetry and 3D Imaging 2nd edition De Gruyter Berlin Germany 684 p.*

*Magnus J.R. and H. Neudecker 2019. Matrix differential calculus with applications in statistics and econometrics 3rd edition John Wiley New York 470 p.*

*Menna F. E. Nocerino P. Drap F. Remondino A. Murtiyoso P. Grussenmeyer and N. Börlin 2018. Improving underwater accuracy by empirical weighting of image observations, International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences XLII(2):699-705. DOI: 10.5194/isprs-archives-XLII-2-699-2018.*

*Mikhail E.M. J.S. Bethel and J.C. McGlone 2001. Introduction to Modern Photogrammetry John Wiley New York 496 p.*

*Moniwa H. 1981. The concept of photo-variant self-calibration and its application in block adjustment with bundles, Photogrammetria 36(1):11-29. DOI: 10.1016/0031-8663(81)90006-5.*

*Murtiyoso A. P. Grussenmeyer and N. Börlin 2017. Reprocessing close range terrestrial and UAV photogrammetric projects with the DBAT toolbox for independent verification and quality control, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII(2/W8):171-177. DOI: 10.5194/isprs-archives-XLII-2-W8-171-2017.*

*Murtiyoso A. P. Grussenmeyer N. Börlin J. Vandermeerschen and T. Freville 2018. Open source and independent methods for bundle adjustment assessment in close-range UAV photogrammetry, Drones 2(1):3. DOI: 10.3390/drones2010003.*

*Nocerino E. F. Menna R. Remondino J.-A. Eraldin L. Cournoyer and G. Reain 2016. Experiments on calibrating tilt-shift lenses for close-range photogrammetry, International Archives of Photogrammetry, Remote Sensing, and Spatial Information Sciences XLI(B5):99-105. DOI: 10.5194/isprsarchives-XLI-B5-99-2016.*

*Piponi D. 2004. Automatic differentiation, C++ templates, and photogrammetry, Journal of Graphics Tools 9(4):41-55. DOI: 10.1080/10867651.2004.10504901.*

*Rabbani T. S. Dijkman F. van den Heuvel and G. Vosselman 2007. An integrated approach for modelling and global registration of point clouds, ISPRS Journal of Photogrammetry and Remote Sensing 61(6):355-370. DOI: 10.1016/j.isprsjprs.2006.09.006.*

*Ray S.F. 2002. Applied photographic optics 3rd edition Focal Press London U.K. 680 p.*

*Rupnik E. J. Jansa and N. Pfeifer 2015. Sinusoidal wave estimation using photogrammetry and short video sequences, Sensors 15(12):30784-30809. DOI: 10.3390/s151229828.*

*Tecklenburg W. T. Luhmann and H. Hastedt 2001. Camera modelling with image-variant parameters and finite elements Optical 3-D measurement techniques V (A. Gruen and H. Kahmen editors) Wichmann Verlag Heidelberg Germany pp. 328-335.*

*Triggs B. P. McLauchlan R. Hartley and A. Fitzgibbon 2000. Bundle adjustment - A modern synthesis Vision Algorithms: Theory and Practice, Proceedings of the International Workshop on Vision Algorithms (B. Triggs A. Zisserman and R. Szeliski editors) Springer Berlin Germany pp. 298-372. DOI: 10.1007/3-540-44480-7_21.*

*Tsai R.Y. 1987. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses, IEEE Journal of Robotics and Automation 3(4):323-344.*

*Wolf P.R. and B.A. Dewitt 2000. Elements of Photogrammetry 3rd edition McGraw-Hill New York USA 608 p.*

*Zhang Z. 2000. A flexible new technique for camera calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence 22(11):1330-1334. DOI: 10.1109/34.888718.*

# 7 Algorithms

*Algorithm 1 The pinhole projection (5) is evaluated by a sequence of function calls.*

**procedure** PinholeNoJac($p$, $p_0$, $R$, $c$, $u_0$)

- $a_1 \leftarrow T_3(p, -p_0)$
- $a_2 \leftarrow L(R^T, a_1)$
- $a_3 \leftarrow H(a_2)$
- $a_4 \leftarrow S(c, a_3)$
- $a_5 \leftarrow T_2(a_4, u_0)$
- **return** $a_5$

**end procedure**

*Algorithm 2: The pinhole projection and one of its Jacobians computed by the same sequence of function calls as in Algorithm 1. Each called function computes both the function value and the necessary Jacobians. The complete analytical Jacobian $J$ is formed by multiplying the component Jacobians.*

**procedure** PinholeWithJac($p$, $p_0$, $R$, $c$, $u_0$)

- $(a_1, J_p) \leftarrow T_3(p, -p_0)$, $J_p = \left[\frac{dT_3}{dp}\right]$,
- $(a_2, J_v) \leftarrow L(R^T, a_1)$, $J_v = \left[\frac{dL}{dv}\right]_{v=a_1}$,
- $(a_3, J_q) \leftarrow H(a_2)$, $J_q = \left[\frac{dH}{dq}\right]_{q=a_2}$,
- $(a_4, J_m) \leftarrow S(c, a_3)$, $J_m = \left[\frac{dS}{dm}\right]_{m=a_3}$,
- $(a_5, J_n) \leftarrow T_2(a_4, u_0)$, $J_n = \left[\frac{dT_2}{dn}\right]_{n=a_4}$,
- $J \leftarrow J_n J_m J_q J_v J_p$, $J = \left[\frac{dg}{dp}\right]$,
- **return** $(a_5, J)$

**end procedure**

# 8 Figures

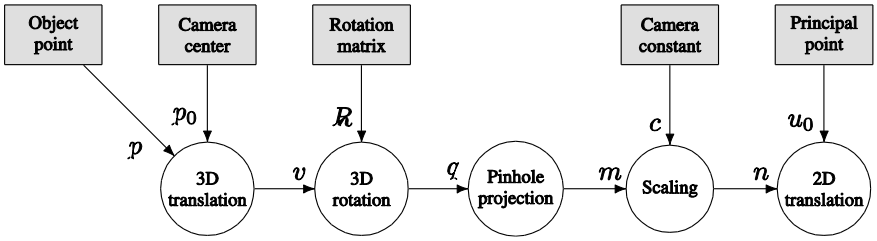


*Figure 1: The computational chain corresponding to equation (5). The grey rectangles indicate bundle adjustment parameters. The white circles indicate component functions from equation (4). The arrows indicate how the parameters and results are propagated. The arrow labels indicate the name of the formal (input) parameter of the corresponding function.*

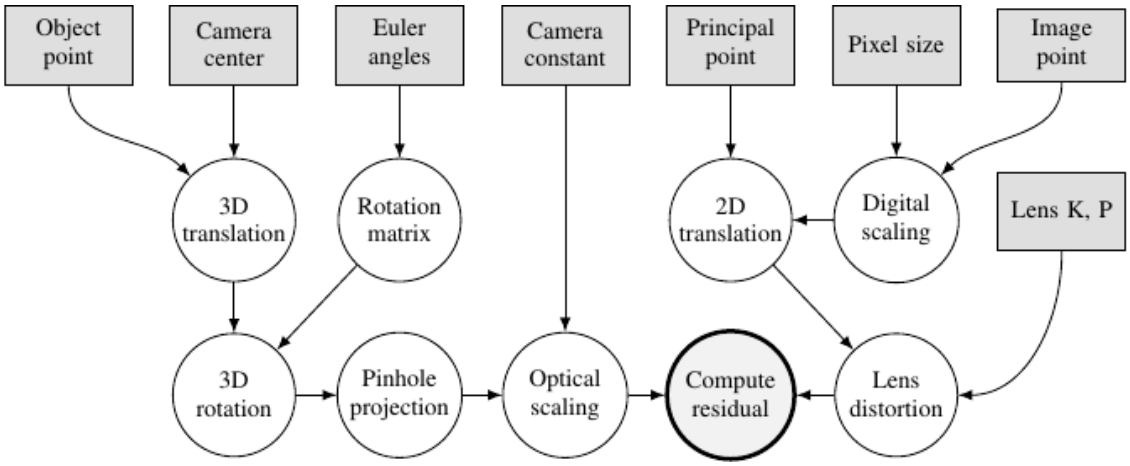


*Figure 2: The computational chain implemented in DBAT. In the photogrammetric formulation, the optical scaling in the camera results in image coordinates expressed in physical units, e.g., mm. The image coordinates are scaled from pixels to mm before the (Brown D. C., 1971) polynomials are used to "correct" the measured image coordinates for lens distortion. The residual (thick circle) is computed as the difference between the projected ideal point and the corrected point.*

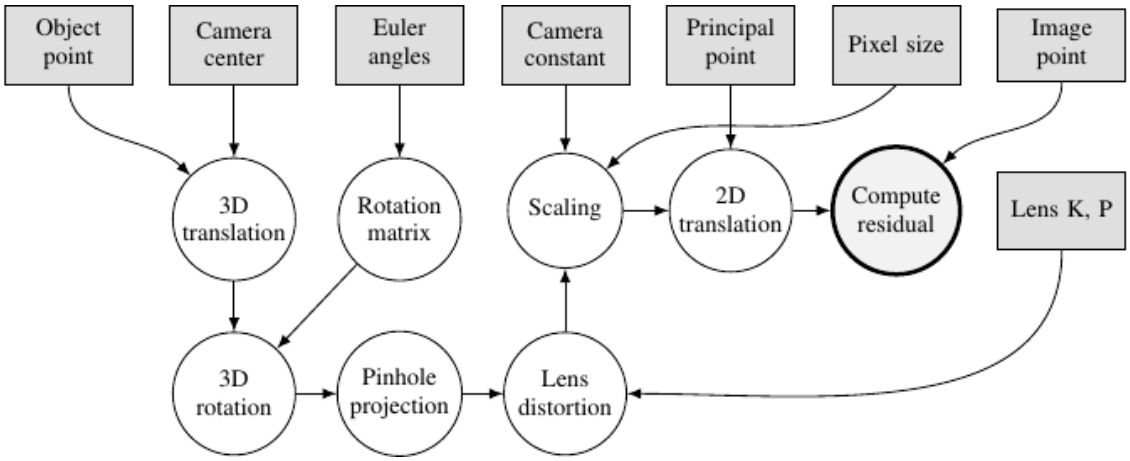


*Figure 3: The DBAT implementation of the Computer Vision formulation of the (Brown D. C., 1971) lens distortion model. The Brown polynomials are used to introduce lens distortion to the projected ideal point in normalized units before the coordinates are scaled directly to pixels. The residual (thick circle) is computed between the measured point and the "distorted" projected point. The same modules have been used as in Figure 2.*

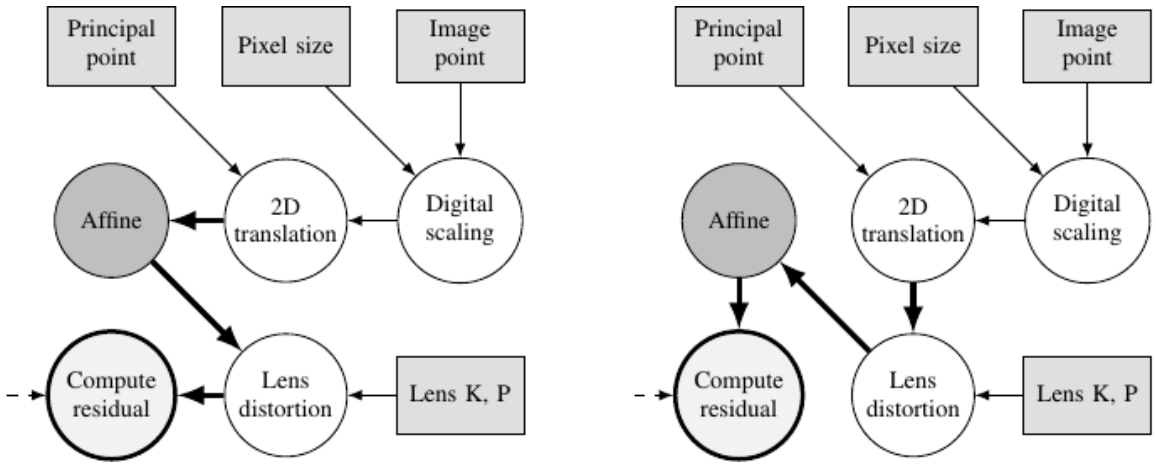


*Figure 4: Two versions of the right-hand side of Figure 2 with different relative placement of the affine transformation with respect to lens distortion. In Model 3 (left), the affine distortion was applied before lens distortion removal. In Model 4 (right), the affine distortion was applied after lens distortion removal. In both cases, the digital scaling uses square pixel sizes. Any non-unit aspect ratio is handled by the affine step.*


*Figure 5: The Nikon D750 DSLR camera with the PC-E Micro NIKKOR 45 mm f/2.8D ED tilt-shift lens used in this paper.*

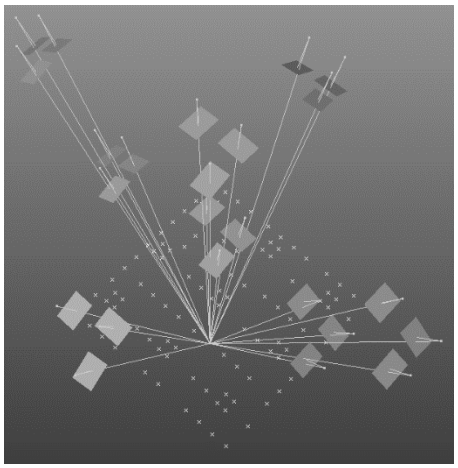

*Figure 6: The synthetic network consisted of 24 cameras at varying roll angles. The simulated targets were in a 3D configuration with a largest dimension of 1000 mm.*

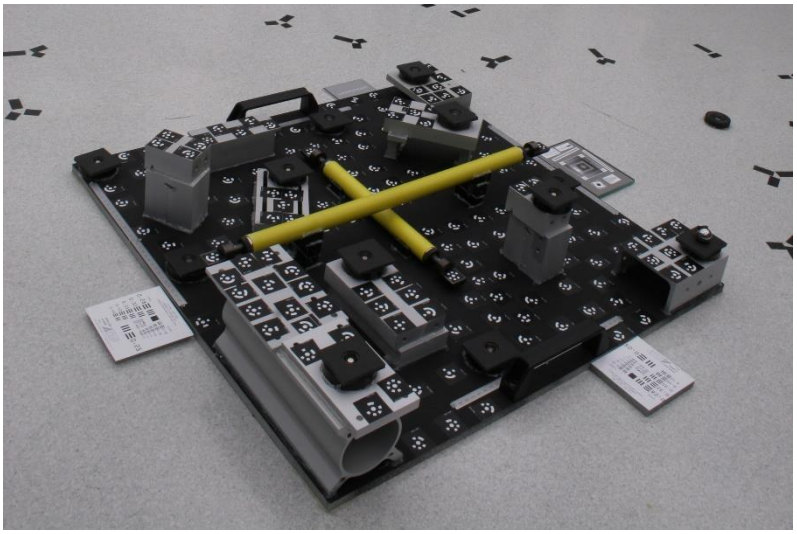

*Figure 7: The 3D test object with about 170 coded targets had a largest dimension of 900 mm.*

# 9 Tables

*Table 1: Assessment of the analysis of the synthetic data sets. The IO parameters columns indicate what IO parameters and how many ($n$) were included in the estimation. The RMS is the point residual over all image observations. The RMSE is the 3D error over all object points. The $\widehat{b_1}$ column shows the estimated $b_1$ value. The rows with the smallest residuals are indicated in grey.*

| Simulated Dataset | DBAT model | IO parameters | | | Internal/external assessment | | |
|---|---|---|---|---|---|---|---|
| | | parameters | $n$ | $\widehat{b_1}$ | $\widehat{\sigma_0}$ | RMS (px) | RMSE (µm) |
| 3 | 2 | $P$ | 8 | 0 | 34.2847 | 4.56 | 620.60 |
| | 3 | $b_1 b_2 P$ | 10 | 0.01218 | 0.0003 | 0.00 | 0.01 |
| | 4 | $b_1 b_2 P$ | 10 | 0.01223 | 0.3773 | 0.05 | 8.26 |
| 4 | 2 | $P$ | 8 | 0 | 34.1330 | 4.54 | 616.59 |
| | 3 | $b_1 b_2 P$ | 10 | 0.01212 | 0.3697 | 0.05 | 8.33 |
| | 4 | $b_1 b_2 P$ | 10 | 0.01218 | 0.0003 | 0.00 | 0.01 |

*Table 2: Assessment of DBAT estimation models 2, 3, and 4 on the TILTED data set. The IO parameters and assessment are as in Table 1, except that the target coordinates computed from the NORMAL data set were used as the true data. The redundancy (number of observation minus number of parameters) was around 7800. The smallest residuals were obtained when $b_1$ and P were estimated (gray rows).*

| DBAT Model | IO parameters | | | | Internal/external assessment | | |
|---|---|---|---|---|---|---|---|
| | $b_1$ | $b_2$ | $P$ | $n$ | $\sigma_0$ | RMS (px) | RMSE (µm) |
| 2 | | | $P$ | 8 | 5.9 | 0.79 | 82.3 |
| | | | | 6 | 5.9 | 0.80 | 92.9 |
| 3 | $b_1$ | | $P$ | 9 | 1.1 | 0.15 | 11.9 |
| | $b_1$ | $b_2$ | $P$ | 10 | 1.1 | 0.15 | 11.9 |
| | $b_1$ | | | 7 | 3.0 | 0.41 | 72.9 |
| | $b_1$ | $b_2$ | | 8 | 3.0 | 0.41 | 73.2 |
| | | $b_2$ | | 7 | 5.9 | 0.80 | 92.6 |
| | | $b_2$ | $P$ | 9 | 5.9 | 0.79 | 82.3 |
| 4 | $b_1$ | | $P$ | 9 | 1.1 | 0.15 | 11.9 |
| | $b_1$ | $b_2$ | $P$ | 10 | 1.1 | 0.15 | 11.9 |
| | $b_1$ | | | 7 | 3.0 | 0.41 | 73.3 |
| | $b_1$ | $b_2$ | | 8 | 3.0 | 0.41 | 73.3 |
| | | $b_2$ | $P$ | 9 | 5.9 | 0.79 | 82.3 |
| | | $b_2$ | | 7 | 5.9 | 0.80 | 92.8 |

*Table 3: Results of the image-variant experiments for DBAT model 3 without $b_2$ with three different sets of image-variant parameters. The number of IO parameters increase from $n = 9$ for the fully block-invariant case to $n = 432$ for the fully image-variant case. Since a corresponding number of prior observations are added, the redundancy remains around 7800. The grey row corresponds to the first model 3 row of Table 2.*

| IO parameters | | Internal/external assessment | | |
|---|---|---|---|---|
| Image-variant params | $n$ | $\widehat{\sigma_0}$ | RMS (px) | RMSE (µm) |
| None | 9 | 1.1 | 0.15 | 11.9 |
| $c, x_0, y_0$ | 150 | 1.0 | 0.14 | 12.0 |
| $c, x_0, y_0, K_i, P_i$ | 385 | 1.0 | 0.13 | 12.1 |
| $c, x_0, y_0, K_i, P_i, b_1$ | 423 | 1.0 | 0.13 | 12.1 |